Theory P2

Thursday, July 8, 2021 8:09 PM

DATATYPES

1. INTEGER : (0-9) without decimal e.g 95 94 93 21
2. REAL : (0-9) with decimal e.g 95.1 101.0 9.3 1.5
3. STRING : Alphabets, Numbers, Symbols
4. CHAR : Single character : "P" "p" "A"
5. BOOLEAN : 2 comparison True False

Variable	Value
Mark	60
Subject	"Computer Science"
Grade	'B'
CourseCompleted	TRUE
AverageMark	49.5

(ii) Programming languages support different data types.

Give an appropriate data type for each of these variables from **part (b)(i)**.

Variable	Data type
Mark	INTEGER
Subject	STRING
Grade	CHAR
CourseCompleted	BOOLEAN
AverageMark	REAL

[5] 5/5

List of variable DATA IDENTIFIER TABLE

List of variable

DATA IDENTIFIER TABLE

LOV	Description	Datatype
Projectcompleted	It will tell the status of the project (True or False)	Boolean

(What an "Identifier table" contains ?)

- List of variable names
- Explanation of what that variable does
- Datatype of that variable

ALGORITHM

- Sequence of steps / instructions
- To implement a task / solution to a problem

STAGES OF ALGORITHMS

① Input : Input N1, N2

② Process : N1 + N2 = Answer

③ Output Print Answer

A software developer is writing a program and includes several features to make it easier to read and understand.

1. Indentation ✓
2. Blank lines ✓
3. Capitalization of keywords ✓
4. Sensible Variable Name ✓
5. Comments ✓
6. Prettyprint ✓

Source Code: It represents a solution / design / algorithm expressed in High level language.

Object Code:
- It is produced by the compiler during translation stage
- The object code is produced by translating the source code.
- Not produced by interpreter.

Pseudocodes: It is a way of using keywords and identifiers to describe an algorithm without following the syntax of a particular programming language.

Flowchart: Pictorial or graphical representation of a program in a flow.

Four Basic Type Of Constructs.

→ Building Blocks of programming

(1) Assignment

A value is given a name (identifier) or the value associated with a given identifier is changed.

Explanation.

Flag = True Number = 5

Number = Number - 1

Number = 4

② SEQUENCE

A number of steps are performed one after the other

Explanation.

③ SELECTION

Under certain conditions some of the steps are performed, otherwise no or different steps are performed.

IF ... THEN ... ELSE ENDIF ✓

CASE OF otherwise EndCase ✓

④ REPITITIONS / ITERATIONS:

- A sequence of steps is performed a number of times.
- This is also known as Looping

(1) For ... To ... Next (count-controlled loop) ✓

(2) While ... Do ... Endwhile (Pre conditional loop) ✓

(3) Repeat ... until (Post-conditional loop) ✓

Statement	Selection	Repetition (Iteration)	Assignment
WHILE Count < 20		✓	
Count ← Count + 1			✓
IF MyGrade <> 'C' THEN	✓		
Mark[Count] ← GetMark(StudentID)			✓
ELSE OUTPUT "Fail"	✓		
ENDFOR		✓	

[6] 6/6

IDENTIFIER

① Constant

- Fix not changeable

② Variable

temporary memory

③ Arrays

temporary memory.

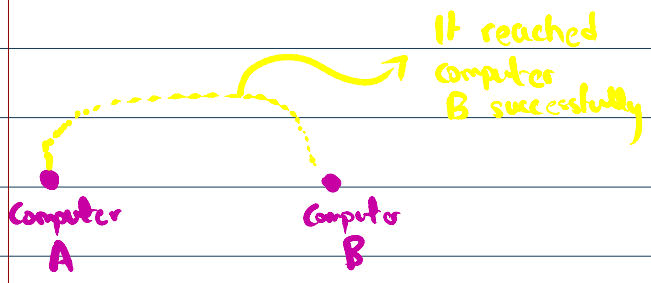
Constant : the value can not be changed accidently during the execution of program

Variable : Stores a value that can change during execution

Arrays : A list of items of the same data type stored under a single name.

TEST DATA

VERIFICATION



- Checking that data has not changed during transfer / input to a compute.

- Double Entry
- Parity Check
- Checksam.

VALIDATION.

- checking that data meets certain criteria

- Range check
- Presence check
- Length check
- Format check
- Character check
- Type check

TYPES OF TEST DATA

① NORMAL DATA ✓

$5 \leq x \leq 10$

Within range (accepted) ✓

-2 12 14

② ABNORMAL DATA

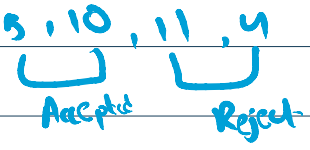
out of the range (rejected) 5 10

③ Extreme

In range, largest and smallest value which is accepted.

(4) Boundary

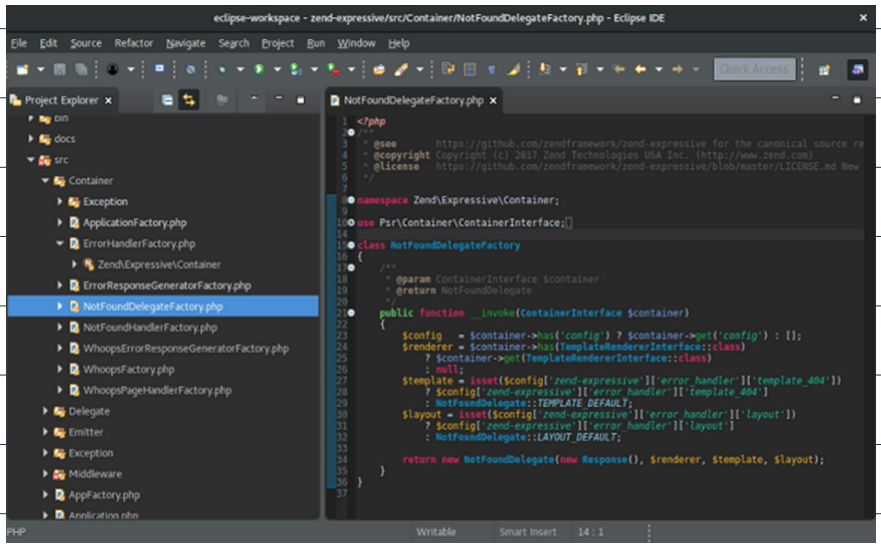
At edge, Accepted, Rejected.



What are the features of a program

- Function / Procedure
- Sequence ✓
- Iteration ✓
- Selection ✓
- Parameters
- Variable ✓
- Logic Operation

(P1) INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)



Integrated development Environment is a software application that combines all of the features and tools needed by a software developer

What are the features of an IDE ?

- Pretty Print ✓
- Automatic Indentation ✓
- Syntax checking ✓
- HighLights any undeclared variable ✓
- Type checking.

What are the Features provided by an IDE that assist in "Initial Error Detection"

- Dynamic Syntax Checking
- Type checking
- Identification of unused variable.

(P1) Describe methods you could use to find errors ?

- Dry run, tracetable
- Breakpoint : Run the code to set point to find error
- Variable watch: Checks the content of variable at specific points
- Stepping : Execute the code line by line.

Program Development Cycle

6 stages

Problem Solving
Design
Coding
Translating
Execution
Testing

5 stages

✓ Analysis
✓ Design
✓ Coding
✓ Testing
✓ Maintenance

EDITOR : Produces the source code in high level language

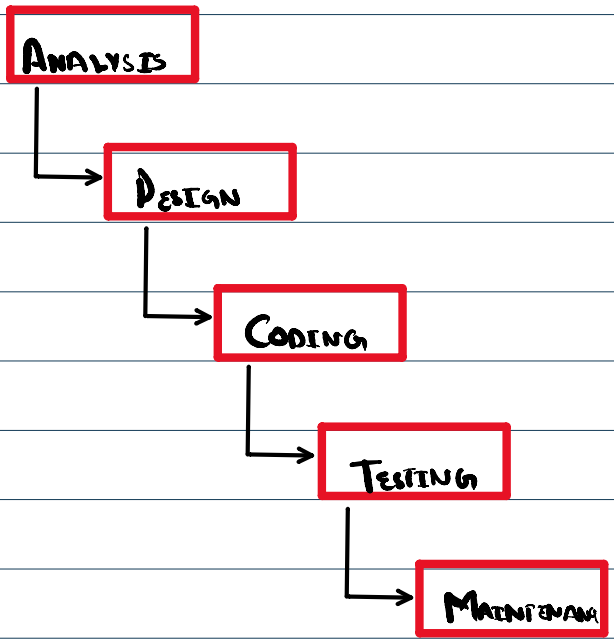
TRANSLATOR : converts source code into object code.

DEBUGGER : A debugger is used to test the program / detect errors.

Development Model

Waterfall Model | Iterative Model | Rapid Application Development

Waterfall Method



Principles
• linear, as each stage is completed before the next is begun • Well documented • low customer involvement (only at the start and end of the process)

BENEFIT	DRAWBACK
• Easy to manage, understand and use • stages do not overlap and are completed on at a time • Works well for smaller programs.	• Difficult to change the requirement at a later stage • Subject to change programs are not suitable. • Working program is produced late in the life cycle. • Not suitable for complex projects

ITERATIVE MODEL

First Waterfall ~~> if successful then moves further

Extended Waterfall

Further extended

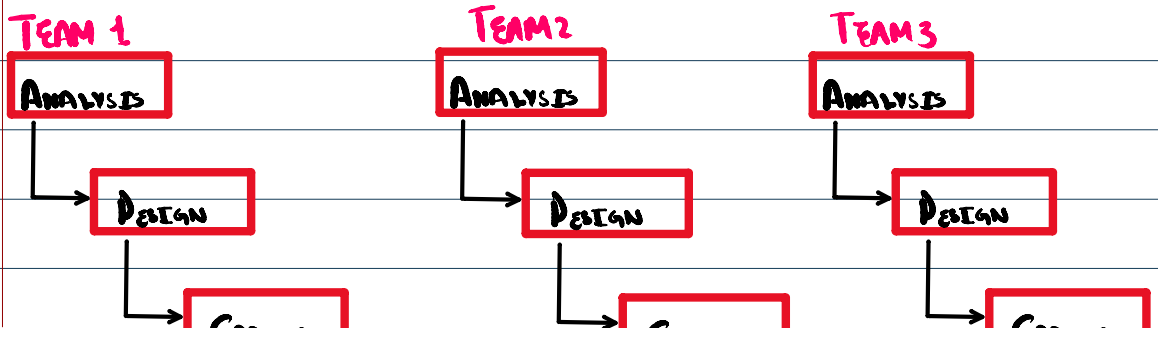
Extends the development if going successful

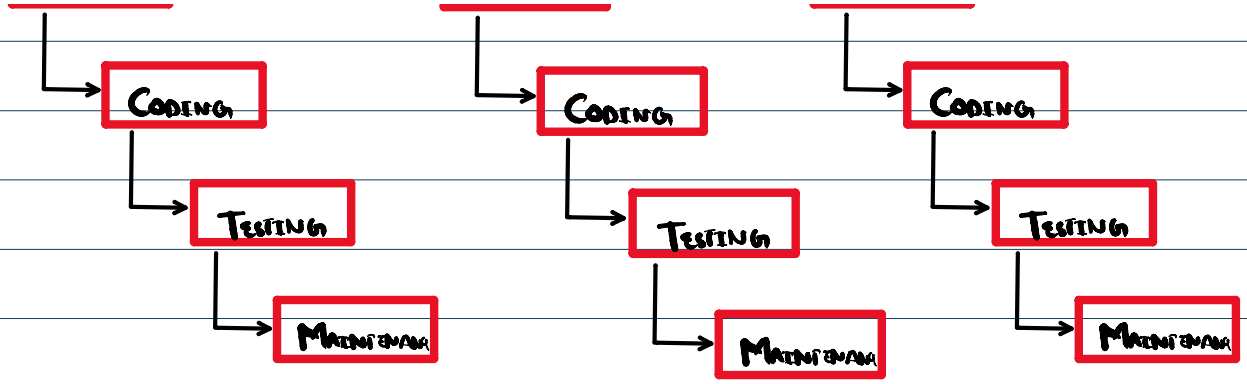
→ Final waterfall.

Principles	.Incremental development as the program development lifecycle is repeated. • High customer involvement as part of the system can be shown to the customer after every iteration.

BENEFIT	DRAWBACK
• Easier to test and debug • more flexible as easier to alter requirements • Customer involved at each iteration therefore no surprises when final system delivered.	• Whole system needs to be defined at start • Needs a good planning overall and at every stage. • Not suitable for short simple projects

RAPID APPLICATION DEVELOPMENT.





. Program is divided into modules and each module has one team and its own waterfall.

Principles	• Minimal Planning • Uses previously written code where possible • high customer involvement.

BENEFIT	DRAWBACK
• Reduced overall development time • Rapid feedback informs the development. • Very flexible as requirement evolve from feedback. • Modification is easier because	• System under development needs to be modular • need strong team of skilled developers, • not suitable for short-simple projects.

- Modification is easier because each part must work independently.

Transferrable Skills

There are certain skills that a person has in a certain language those can be used in another language.

Explaination.

You should be able to recognize / understand in another language :-

- Declaration
- Assignment
- Sequence
- Selection
- Repition/ Iteration
- Subroutines
- Parameters passed between module
- Input output

Exam Style Question:

An algorithm is implemented in a high-level language. Changes are required and the program is given to Albert, who is an experienced programmer but He is not familiar with the language that has been used.

Explain why Albert would be able to understand the program ?

- He would use his transferable skills to understand the new program.
- He could recognise basic control structures in the language such as loops, selectional statements, declaration.
- He could read the comments / meaningfull variable names.

Types Of Maintenance.

1. Corrective
2. Perfective
3. Adaptive
4. Preventive.

Corrective Maintenance

- Maintenance occurs when testing reveals a fault (or error) in the program and this is corrected.
- Corrective maintenace is when errors are found and fixed.

What are the reason for using corrective maintenance?

- To ammend the algorithms to eliminate errors.

Adaptive Maintenance

- It is a term used for changes that made to a program in response to a specification changes
- As a result of changes to requirements or available technology.

What are the reason for using adaptive maintenance?

In response to spesification change arising from changes to business rules or environments.

Perfective Maintenance

- Perfective Maintenance mainly deals with implementing new or changed user requirement
- Perfective Maintenance involves making functional enhancement
- This includes enhancing both the function and efficiency of the code and changing the functionalities as per the user's changing needs.

Preventive Maintenance.

- Prorentive maintenanc involves performing activities to prevent the occurence of error.

Types Of Error

SYNTAX ERROR | LOGICAL ERROR | RUN-TIME

SYNTAX ERROR.

This is basketball very new

AVERAGE TO DISPLAY (Print)

- Are small gramatical mistakes, sometimes limited to a single character

For e.g : A missing semi-colon at the end of a line or an extra bracket at the end of a function may product a syntax Error.

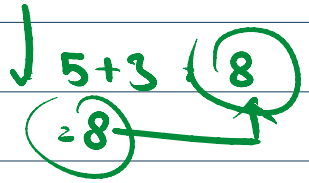
LOGIC ERROR.

- Program does not perform as expected
- Errors in the logic of a program

RUN-TIME ERROR

- Program executes an invalid instruction (out of bounds error)
- Attempts to divide by zero
- Program Crashes.

SOFTWARE TESTING



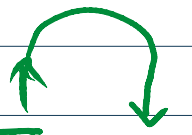
BLACK-BOX TESTING

- Match expected ans to actual ans
- A person does not need to know the structure of the code.
- A person chooses normal, boundary and erroneous data.

White Box - Testing.

- Test every possible 'Logic path' through the code
- A trace table is often used during the program testing.

Alpha - Testing.



- In house testing
- Testing is done by developers in-house other than the developer who actually built the software.

Beta - Testing.

- Out of house testing
- Performed by group of people those are assigned to find out bugs (errors) in the code.

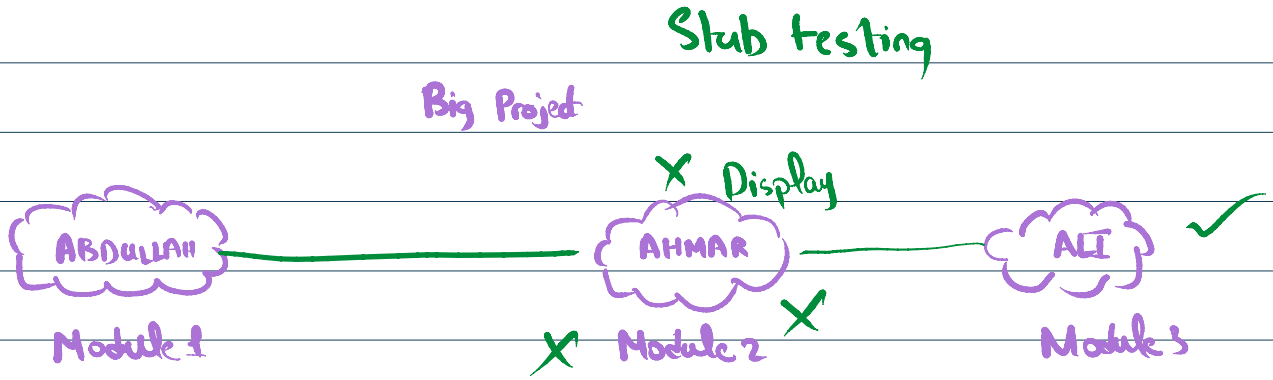
Stub Testing.

- Simple / dummy module written to stimulate the model.
- Test carried out all the modules (subroutines) have been written.

Integration Testing

- Is the testing of any separately written modules to ensure that they work together.

Explanation



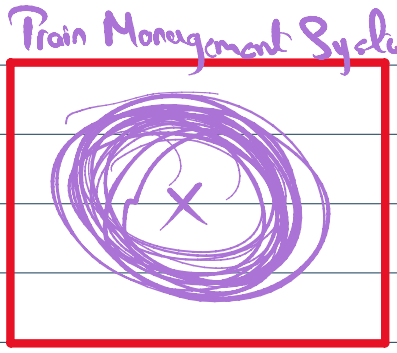
Integration testing = Combining and then testing
If a module is missing
then stub testing.

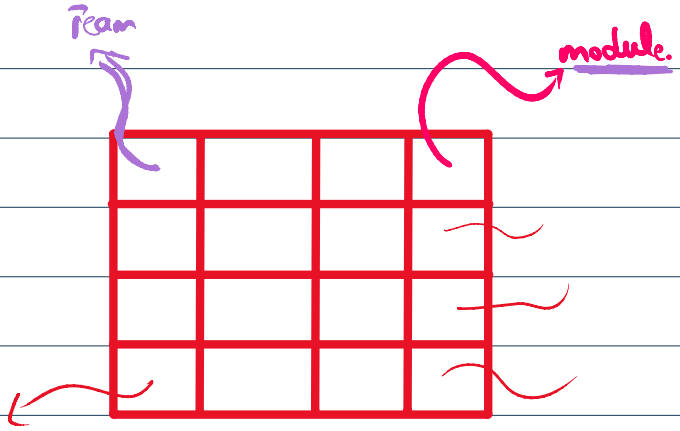
Acceptance Testing:

- Is used for the completed program to prove to the customer/client that it works as required.
- End-user testing.

Stepwise Refinement

Block of Code Explaination





What is stepwise refinement?

- The breaking down of an algorithm
- To a level of sufficient (detail) into smaller part
- From which it can be programmed.

Note: Each sub-task can be considered as a module that is refined seperately

- Module or Subroutine can be either a function or a procedure.

what are the uses of subroutine in a program?

- to perform a frequently used operation within a program
- that can be called when needed
- that can be reused by another program

why it is a good practice to construct the program using modules?

- To make more manageable / understandable solution.
- Subroutine / module may be independently tested.

- Subroutine / module may be independently tested.
- Program easier to maintain.

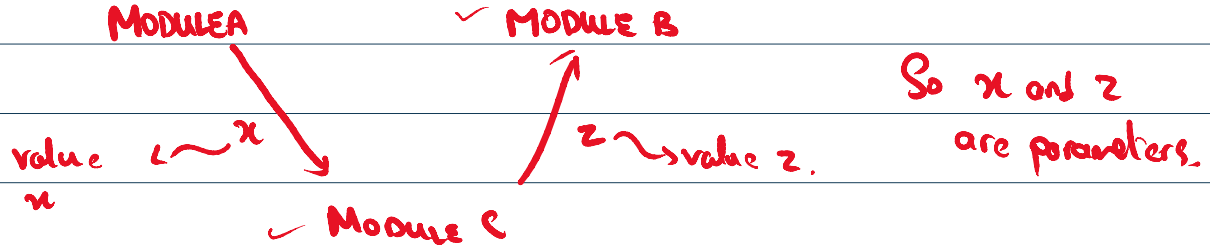
What are the advantages of sub-routine ?

- Allows the subroutine to be called into multiple places. ✓
- Sub-routine may be tested independently. ✓
- If the task changes, the changes needs to be made only once. ✓
- Reduces unnecessary duplications ✓
- Allows team to work on different parts of the solution.

PROCEDURE : A sequence of steps that is given an identifier and can be called to perform a sub-task (IT DOES NOT RETURN A VALUE)

FUNCTION A sequence of steps that is given an identifier and can be called to perform a sub-task (RETURNS A VALUE)

Describe two methods of passing the parameters between modules. ~ Values passed b/w modules



So x and z are parameters.

① By Value: A copy of variable itself is passed and original value is not changed.

② By Reference: The address of the variable is passed original value is changed.

State how structured programming supports the implementation of sub-task?

- Subroutines
- Functions
- Procedures
- Parameters.